Distributed Faulty Node Detection in DTNs in Presence of Byzantine Attack

Wenjie Li[*], Francesca Bassi[*†], Michel Kieffer[*‡§], Alex Calisti[¶], Gianni Pasolini[¶], and Davide Dardari[¶]
[*]Laboratoire des Signaux et Systèmes (L2S, UMR CNRS 8506) CNRS-CentraleSupelec-Université Paris-Sud
3, rue Joliot Curie 91192 Gif-sur-Yvette, France
[†]ESME-Sudria, 94200 Ivry-sur-Seine, France
[‡]LTCI Telecom ParisTech, 75013 Paris, France
[§]Institut Universitaire de France, 75005 Paris, France
[¶]CNIT, DEI, University of Bologna, Italy.

***Abstract*—This paper considers a delay tolerant network consisting of nodes equipped with sensors, some of them producing outliers. A distributed faulty node detection (DFD) algorithm, whose aim is to help each node in estimating the status of its sensors, has been proposed recently by the authors. The aim of this paper is to analyze the robustness of the DFD algorithm to the presence of misbehaving nodes performing Byzantine attacks. Two types of attacks are considered and analyzed, each trying to mislead the other nodes in the estimation of the status of their sensors. This provides insights on the way the parameters of the DFD algorithm should be adapted to minimize the impact of misbehaving nodes. Theoretical results are illustrated with simulations considering nodes with random displacements, as well as traces of node inter-contact times from real databases.**

I. Introduction

Delay Tolerant Networks (DTNs) refer to the challenging situation of networks operating with intermittent connectivity [1]. This happens, for example, in Vehicular DTNs [2], where the nodes are moving vehicles and communication is established only between closely located agents. This produces frequent link disruptions and network topology reconfiguration. This time-varying nature exposes DTNs to infiltrations by potentially malicious nodes, who may attempt to perturb the DTN behavior. Threatens against the DTN integrity may come in the form of malware attacks [3], selfish behavior of nodes [4], Byzantine attacks [5], [6], and so on. The absence of a central unit able to act as a certifying authority makes trust management in DTNs very difficult.

In this paper we consider a DTN where nodes are equipped with sensors, collecting data used, *e.g.*, to estimate some physical phenomenon. We assume that the network behavior is perturbed by nodes with defective sensors and by nodes performing Byzantine attacks.

A sensor is called defective if it frequently reports erroneous measurements. This phenomenon may be due, *e.g.*, to the degradation of the equipment in time. The identification of nodes equipped with defective sensors is very important to save communication resources and to prevent erroneous measurements to pollute the estimates provided by the DTN. Distributed fault detection (DFD) is a well-investigated topic in Wireless Sensor Networks (WSNs), see [7]–[9] and references therein. The WSNs considered in the literature are usually dense and have a static topology. DFD in DTNs is made more challenging by the sparse and dynamic topology, and is much less investigated. The authors proposed in [10] a fully distributed and easily implementable algorithm allowing each node of a DTN to determine whether its own sensors are defective.

找到这些 有问题的node(带有错误传感器的)

A basic assumption in [10] is that all the nodes in the DTN may not be misbehaving in other ways than carrying defective sensors. This paper investigates the performance of the DFD algorithm when the DTN is under Byzantine attack, *i.e.*, several nodes are fully controlled by an adversary. While the normal nodes perform the DFD algorithm to determine the status of their own sensors, the Byzantine nodes try to prevent the correct self-evaluation of normal nodes. This work aims to determine *i*) whether the DFD algorithm proposed in [10] is robust against the introduction of Byzantine nodes; *ii*) how to adjust the algorithm parameters to minimize the effects of the Byzantine attack. To answer these questions we extend the analysis in [10] by taking into account a proportion of Byzantine nodes. Theoretical predictions are supported by simulation results obtained by using both an idealized node displacement model and traces from real databases.

The rest of the paper is organized as follows. Section II presents the system model and the basic assumptions. Section III details the DFD algorithm for DTNs and introduces the Byzantine attack model. Section IV discusses the dynamics of different nodes during the DFD algorithm. Section V analyzes the property of the equilibrium obtained from the state equations and discusses the choice of the parameters in the algorithm. Section VI provides numerical results and Section VII concludes the paper.

II. System Model

Consider a set $\mathcal{S}$ of moving nodes equipped with sensors. Assume that a subset $\mathcal{B} \subset \mathcal{S}$ of these nodes are controlled by an adversary and perform a Byzantine attack to disturb the behavior of the network. The nodes in the set $\mathcal{N} = \mathcal{S} \setminus \mathcal{B}$ are *normal*. Let $\mathcal{D} \subset \mathcal{N}$ denote the subset of nodes which are not malicious but produce *outliers* due to their defective sensors. The *outliers* are measurements having statistical characteristics significantly different from normal measurements provided by

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

good sensors. As a consequence, the *status* of Node i has three possible values $\theta_i(t) \in \Theta = \{0, 1, 2\}$, *i.e.*,

$$\theta_i(t) = \begin{cases} 0, & \text{if } i \in \mathcal{N} \setminus \mathcal{D}, \\ 1, & \text{if } i \in \mathcal{D}, \\ 2, & \text{if } i \in \mathcal{B}. \end{cases}$$

In this paper, one assumes that the status of nodes remains constant during the algorithm, *i.e.*, $\theta_i(t) = \theta_i$, and that the nodes are initially not aware of their status and only nodes in $\mathcal{N}$ are willing to estimate their status. Let p_θ be the proportion of nodes with status $\theta \in \Theta$, with $p_0 + p_1 + p_2 = 1$.

Nodes can exchange information only during the limited time interval in which they are in vicinity. As in [10], nodes are assumed to be well-mixed and the time interval between two successive meetings of a given node is assumed to follow an exponential distribution with an inter-contact rate λ [11]. Moreover, one assumes that each meeting involves only two nodes. When more than two nodes meet at the same time instant, processing is performed pair-by-pair.

During each meeting of a pair of nodes $(i, j) \in \mathcal{S}$ each node senses data $m_{\cdot}$ with their own sensors and then may exchange these data. If Node i has received the data from Node j (*i.e.*, m_j), then a *local outlier detection test* (LODT) can be performed by Node i with outcome y_{ij}. Assume that the spatial and temporal correlation between data is such that only data sensed during the meeting of two nodes can be exploited by a LODT. Therefore, previously collected data are not exploited. The LODT yields $y_{ij} = 1$ if it detects the presence of at least an outlier among the data m_i and m_j, and $y_{ij} = 0$ otherwise. The LODT is not able to determine which sensor is producing outliers. Such situation occurs for example, when comparing few scalar measurements of the same physical quantity. The presence of an outlier is easily detected when the measurements are very different. Nevertheless, even if the difference is large, it is difficult to determine which measurement is an outlier.

LODTs can take various forms, see [9]. In this paper, the LODT is characterized by the probabilities $q_{\theta_i\theta_j} = \mathbb{P}\{Y_{i,j} = 1 \mid \theta_i, \theta_j\}$, with $\theta_i \in \Theta$ and $\theta_j \in \Theta$. For example, q_{00} is the probability that an outlier is detected when data are provided by good sensors. One has $q_{\theta_i\theta_j} = q_{\theta_j\theta_i}$ as $y_{ij} = y_{ji}$. One further assumes that $q_{00} < q_{01} = q_{10} \leqslant q_{11}$, which is reasonable, since the outcome of a LODT is more likely to be 1 as the number of outliers involved increases.

The properties of LODTs when a malicious node is involved will be further discussed in Section III-B.

III. DFD Algorithm Subject to Byzantine Attacks

This section recalls the DFD algorithm presented in [10] and then discusses the behavior of misbehaving nodes.

A. DFD algorithm

In the DFD algorithm [10], each node manages two counters $c_{\mathrm{m},i}(t)$ and $c_{\mathrm{d},i}(t)$ with $c_{\mathrm{m},i}(0) = c_{\mathrm{d},i}(0) = 0$. Using $c_{\mathrm{m},i}(t)$, Node i counts the number of LODTs that it has performed. Using $c_{\mathrm{d},i}(t)$, Node i counts the number of LODTs resulting in the detection of outliers, *i.e.*, $y_{i\cdot} = 1$. Consider ν as a constant decision threshold, Node i sets its own estimate $\widehat{\theta}_i(t) = 1$ if $c_{\mathrm{d},i}(t)/c_{\mathrm{m},i}(t) \geqslant \nu$. Otherwise, it sets $\widehat{\theta}_i(t) = 0$.

Only the nodes with $\widehat{\theta}(t) = 0$ can send their data to the nodes met at time t. Each node performs a LODT and updates its counters only when it has received some data from another node. For example, assume that Node i with $\widehat{\theta}_i(t) = 1$ meets Node j at time t. Node i still takes measurements, but it does not send these data to Node j. If $\widehat{\theta}_j(t) = 0$, then Node i can receive the data from Node j and perform a LODT.

To simplify the analysis, one has chosen to consider the evolution of $c_{\mathrm{m},i}(t)$ and $c_{\mathrm{d},i}(t)$ over a sliding time window containing the time instants of the last M meetings during which Node i has performed a LODT. Algorithm 1 summarizes the proposed DFD technique for an arbitrary normal Node $i \in \mathcal{N}$.

Algorithm 1 Sliding-Window DFD algorithm for Node $i \in \mathcal{N}$.

```
1) Initialize $t_i^0 = 0$, $\widehat{\theta}_i(t_i^0) = 0$, $c_{\mathrm{m},i}(t_i^0) = c_{\mathrm{d},i}(t_i^0) = 0$, $\iota = 1$, and $\mu = 0$.
2) Do $\widehat{\theta}_i(t) = \widehat{\theta}_i(t_i^{\iota-1})$, $c_{\mathrm{m},i}(t) = c_{\mathrm{m},i}(t_i^{\iota-1})$, $c_{\mathrm{d},i}(t) = c_{\mathrm{d},i}(t_i^{\iota-1})$, and $t = t + \delta t$ until the $\iota$-th meeting occurs at time $t_i^\iota$ with Node $j^\iota \in \mathcal{S}$.
3) Perform local measurement of data $m_i(t_i^\iota)$.
4) If $\widehat{\theta}_i(t_i^\iota) = 0$, then transmit $m_i(t_i^\iota)$ to Node $j^\iota$.
5) If data $m_{j^\iota}$ have been received from Node $j^\iota$, then
    a) $\mu = \mu + 1$. Perform a LODT with outcome $y_i^\mu$.
    b) Update $c_{\mathrm{m},i}$ and $c_{\mathrm{d},i}$ as
        $\begin{cases} c_{\mathrm{m},i}(t_i^\iota) = \min\{\mu, M\} \\ c_{\mathrm{d},i}(t_i^\iota) = \sum_{m=\max\{1,\mu-M+1\}}^{\mu} y_i^m \end{cases}$    (1)
    c) Update $\widehat{\theta}_i$ as follows
        $\widehat{\theta}_i(t_i^\iota) = \begin{cases} 1 & \text{if } c_{\mathrm{d},i}(t_i^\iota)/c_{\mathrm{m},i}(t_i^\iota) \geqslant \nu, \\ 0 & \text{else} \end{cases}$    (2)
6) $\iota = \iota + 1$. Go to 2.
```

见证人

B. Byzantine attack

To disturb the behavior of Algorithm 1, a Byzantine Node b may set $\widehat{\theta}_b(t) = 0$, $\forall t \geqslant 0$, so that it always indicates to the encountered nodes that it is well behaving and that it trusts its sensors. Then Node b may transmit some artificial data to mislead the other nodes. Two types of behavior are considered in what follows.

B1) Node b always transmits random quantities to the encountered nodes. These random data are usually outliers. Therefore, q_{20} and q_{21} are close to 1.

B2) Node b performs a measurement m_b and always waits for the data m_i coming from the encountered Node i. If m_i is close to m_b then it is likely that Node i is carrying good sensors. To introduce confusion, Node b does not send m_b, but sends a significantly different quantity to Node i. If m_i is very different from m_b, it is likely that Node i is carrying a defective sensor. To increase confusion, Node b transmits to Node i a quantity similar to m_i. In this case q_{20} is close to 1 and q_{21} is close to 0.

IV. Dynamics of the DFD algorithm under Byzantine attack

Define the triple $\mathbf{x}_i(t) = (\theta_i, c_{\mathrm{m},i}(t), c_{\mathrm{d},i}(t))$ to represent the state of Node $i \in \mathcal{N}$. The evolution of the state of Node i, conditioned by its status θ_i, follows a Markov model. In particular, there are two chains as $\theta \in \{0,1\}$.

In order to simplify the notations, let $c_{\mathrm{m},i}(t) = \ell$ and $c_{\mathrm{d},i}(t) = k$. At time t, among the nodes with status $\theta \in \{0,1\}$, denote $X_\theta^{\ell,k}(t)$ as the proportion of nodes in state $\mathbf{x}(t) = (\theta, \ell, k)$. The state transition probabilities of nodes are evaluated in Section IV-A. Then the evolution of $X_\theta^{\ell,k}(t)$ are described in Section IV-B.

A. Transition probabilities

见证人

Define $\pi_\theta^{\delta_\mathrm{m},\delta_\mathrm{d}}$ as the transition probability from State (θ, ℓ, k) to State $(\theta, \ell+\delta_\mathrm{m}, k+\delta_\mathrm{d})$. In case where $c_{\mathrm{m},i}(t) = \ell < M$, the counter $c_{\mathrm{m},i}(t)$ either increases or remains constant, thus $(\delta_\mathrm{m}, \delta_\mathrm{d}) \in \{(0,0),(1,0),(1,1)\}$. The only possibility leading to $\delta_\mathrm{m} = 0$ is that Node J is not a Byzantine node and $\widehat{\theta}_J(t) = 1$. Therefore, for any $\theta \in \{0,1\}$,

$$\pi_\theta^{0,0}(t,\ell,k) = \sum_{\theta \in \{0,1\}} \mathbb{P}\{\theta_J = \theta\}\, \mathbb{P}\left\{\widehat{\theta}_J(t) = 1 | \theta_J = \theta\right\}$$

delta_{m} 和 delta_{d}都为0， 仅当遇到Node J(自己认为自己是1)

$$= p_0 p^{01}(t) + p_1 p^{11}(t), \tag{3}$$

where $p_\theta = \mathbb{P}\{\theta_J = \theta\}$ by the assumption that the nodes are well mixed. One introduces

Node J自己是0 估计成1；本来是1 估计成1

$$p^{\theta\widehat{\theta}}(t) = \mathbb{P}\left\{\widehat{\theta}_J(t) = \widehat{\theta} | \theta_J = \theta\right\}, \tag{4}$$

which is the proportion of agents with status θ believing their status is $\widehat{\theta}$. Notice that $p^{\theta\widehat{\theta}}(t)$ can be obtained from $X_\theta^{\ell,k}(t)$ according to the decision rule (2), *i.e.*,

$$\begin{cases} p^{\theta 0}(t) & = X_\theta^{0,0}(t) + \sum_{\ell,k: k/\ell<\nu} X_\theta^{\ell,k}(t). \\ p^{\theta 1}(t) & = \sum_{\ell,k: k/\ell \geqslant \nu} X_\theta^{\ell,k}(t). \end{cases} \tag{5}$$

A state transition occurs with $(\delta_\mathrm{m}, \delta_\mathrm{d}) = (1,1)$ when Node i with status $\theta_i = \theta$ meets Node J with $\widehat{\theta}_J(t) = 0$ *and* when the LODT yields $y_i(t) = 1$. The two events are independent, hence

$$\pi_\theta^{1,1}(t,\ell,k) = \sum_{\phi \in \Theta} \mathbb{P}\left\{Y_{iJ}(t) = 1, \theta_J = \phi, \widehat{\theta}_J(t) = 0 | \theta_i = \theta\right\}$$

$$= \sum_{\phi \in \Theta} \mathbb{P}\{\theta_J = \phi\}\, \mathbb{P}\left\{\widehat{\theta}_J(t) = 0 | \theta_J = \theta\right\}.$$

$$\cdot\, \mathbb{P}\{Y_i(t) = 1 | \theta_i = \theta, \theta_J = \phi\} = \sum_{\phi \in \Theta} p_\phi q_{\theta\phi} p^{\phi 0}(t). \tag{6}$$

Since the Byzantine nodes with $\theta_b = 2$ always indicate $\widehat{\theta}_b = 0$, one may rewrite (6) as

$$\pi_\theta^{1,1}(t,\ell,k) = p_2 q_{\theta 2} + \sum_{\phi \in \{0,1\}} p_\phi q_{\theta\phi} p^{\phi 0}(t). \tag{7}$$

Finally, $\pi_\theta^{1,0}(t,\ell,k) = \mathbb{P}\left\{Y_i(t) = 0, \widehat{\theta}_J(t) = 0 | \theta_i = \theta\right\}$ is obtained similarly from (6)

$$\pi_\theta^{1,0}(t,\ell,k) = p_2(1-q_{\theta 2}) + \sum_{\phi \in \{0,1\}} p_\phi (1-q_{\theta\phi}) p^{\phi 0}(t). \tag{8}$$

见证人

In the case where $c_{\mathrm{m},i}(t) = M$, one has $\delta_\mathrm{m} = 0$ as the counter $c_{\mathrm{m},i}(t)$ reaches its maximum value. In Algorithm 1, μ is the number of LODTs performed by Node i up to time t. When $\mu \geqslant M$, only the last M LODT outcomes are considered: LODT outcomes y_i^m with $m \leqslant \mu - M$ are discarded. Consider the random event $\mathcal{E}_y(t) = \left\{Y_i^{\mu-M} = y \mid \sum_{m=\mu-M}^{\mu-1} Y_i^m = k\right\}$ in which $y \in \{0,1\}$. This event represents a situation where one knows that k LODTs were positive among the last M tests and the old LODT outcome that will be discarded once the new LODT outcome is available, also concluded in the presence of defective sensors. As discussed in [10], one has $\mathbb{P}\{\mathcal{E}_1(t)\} \approx k/M$ and $\mathbb{P}\{\mathcal{E}_0(t)\} \approx 1 - k/M$.

Assume that the $(\mu - M)$-th LODT performed by Node i occurred at time $\tilde{t}$, then $y_i^{\mu-M}$ can also be denoted as $y_i(\tilde{t})$ and $\delta_\mathrm{d} = y_i(t) - y_i(\tilde{t}) \in \{-1,0,1\}$.

To have $(\delta_\mathrm{m}, \delta_\mathrm{d}) = (0,1)$, three independent events have to occur: 1) the encountered Node J has $\widehat{\theta}_J(t) = 0$; 2) $y_i(t) = 1$; 3) $y_i(\tilde{t}) = 0$, *i.e.*, $\mathcal{E}_0(t)$. The transition probability is then deduced using derivations similar to (6),

$$\pi_\theta^{0,1}(t,M,k) = \frac{M-k}{M}\left(p_2 q_{\theta 2} + \sum_{\phi \in \{0,1\}} p_\phi q_{\theta\phi} p^{\phi 0}(t)\right). \tag{9}$$

Consider then $(\delta_\mathrm{m}, \delta_\mathrm{d}) = (0,-1)$, similarly, one obtains,

$$\pi_\theta^{0,-1}(t,M,k)$$

$$= \frac{k}{M}\left(p_2(1-q_{\theta 2}) + \sum_{\phi \in \{0,1\}} p_\phi (1-q_{\theta\phi}) p^{\phi 0}(t)\right). \tag{10}$$

Considering the last transition $(\delta_\mathrm{m}, \delta_\mathrm{d}) = (0,0)$. To obtain the expression of $\pi_\theta^{0,0}(t,M,k)$, one needs to introduce (9-10) into $\pi_\theta^{0,0}(t,M,k) = 1 - \pi_\theta^{0,1}(t,M,k) - \pi_\theta^{0,-1}(t,M,k)$.

B. Macroscopic evolution

With the transition probabilities discussed in Section IV-A and the initial conditions

$$X_\theta^{0,0}(0) = 1, \text{ and } X_\theta^{\ell,k}(0) = 0, \forall \ell, k \neq 0,$$

the evolution of the various proportions $X_\theta^{\ell,k}(t)$ of nodes in the corresponding states can be obtained, see [10] for the detail. To lighten the equations, consider the function

$$Z_\theta^{\delta_\mathrm{m},\delta_\mathrm{d}}(\ell,k,t) = \begin{cases} X_\theta^{\ell,k}(t)\, \pi_\theta^{\delta_\mathrm{m},\delta_\mathrm{d}}(\ell,k), & \text{if } 0 \leqslant k \leqslant \ell \leqslant M, \\ 0, & \text{otherwise,} \end{cases}$$

then for any $\theta \in \{0,1\}$, one has

$$\begin{cases} \frac{dX_\theta^{\ell,k}}{dt} \overset{(a)}{=} \lambda \sum\limits_{\delta_\mathrm{d} \in \{0,1\}} \left(Z_\theta^{1,\delta_\mathrm{d}}(\ell-1, k-\delta_\mathrm{d}, t) - Z_\theta^{1,\delta_\mathrm{d}}(\ell,k,t)\right) \\ \frac{dX_\theta^{M,k}}{dt} \overset{(b)}{=} \lambda \sum\limits_{\delta_\mathrm{d} \in \{-1,1\}} \left(Z_\theta^{0,\delta_\mathrm{d}}(M, k-\delta_\mathrm{d}, t) - Z_\theta^{0,\delta_\mathrm{d}}(M,k,t)\right) \\ \qquad\qquad + \lambda \sum\limits_{\delta_\mathrm{d} \in \{0,1\}} Z_\theta^{1,\delta_\mathrm{d}}(M-1, k-\delta_\mathrm{d}, t), \end{cases} \tag{11}$$

where (a) describes the evolution of the proportion of state components in the transient regime and (b) is for the permanent regime.

V. Analysis of the Equilibrium

In this section, we investigate the asymptotic behavior of the DTN state equations (11). Algorithm 1 may drive $X_\theta^{\ell,k}$ to an equilibrium $\overline{X}_\theta^{\ell,k}$ at which the proportions of nodes in different states $X_\theta^{\ell,k}(t)$ do not vary any more. As a consequence, $p^{\theta 0}(t)$ defined in (5) also tends to an equilibrium $\overline{p}^{\theta 0}$.

A. Equilibrium of $X_\theta^{\ell,k}$

The results presented in this section are the extension of those in [10] by considering the affect of Byzantine attack.

Proposition 1. *Assume that the dynamic system described by (11) admits some equilibrium $\overline{X}_\theta^{\ell,k}$, then $\overline{\mathbf{p}} = (\overline{p}^{00}, \overline{p}^{10})$ is the solution of (14) (at the top of the next page) and for any $\theta \in \{0,1\}$ and $k \leqslant \ell$,*

$$\overline{X}_\theta^{\ell,k} = \begin{cases} 0, & \forall \ell < M, \\ \binom{M}{k} \left(h_\theta\left(\overline{\mathbf{p}}\right)\right)^k \left(1-h_\theta\left(\overline{\mathbf{p}}\right)\right)^{M-k}, & \ell = M, \end{cases} \tag{12}$$

where

$$h_\theta\left(\overline{\mathbf{p}}\right) = \frac{p_0 q_{\theta 0}\overline{p}^{00} + p_1 q_{\theta 1}\overline{p}^{10} + p_2 q_{\theta 2}}{p_0\overline{p}^{00} + p_1\overline{p}^{10} + p_2}. \tag{13}$$

Proposition (1) can be proved using derivations similar to those presented in [10]. Proposition (1) provides non-linear equations (14) that have to be satisfied by $\overline{\mathbf{p}}$. With the solutions of (14), the values of $\overline{X}_\theta^{M,k}$ at equilibrium can be easily deduced.

B. Approximations of the Equilibrium

Closed-form expressions for $\overline{p}^{00}$ and $\overline{p}^{10}$ are difficult to obtain from (14). This section introduces an approximation of (14) from which some insights may be obtained on the way ν should be chosen to minimize the impact of the presence of misbehaving nodes.

Since both $\overline{p}^{10}$ and $\overline{p}^{01}$ represent the proportions of nodes having wrong estimates of their status, the values of $\overline{p}^{10}$ and $\overline{p}^{01}$ should be small. Thus one may consider the following approximations

$$\begin{aligned} \widetilde{h}_\theta &= \lim_{(\overline{p}^{00},\overline{p}^{10})\to(1,0)} \frac{p_0 q_{\theta 0}\overline{p}^{00} + p_1 q_{\theta 1}\overline{p}^{10} + p_2 q_{\theta 2}}{p_0\overline{p}^{00} + p_1\overline{p}^{10} + p_2} \\ &= \frac{p_0 q_{\theta 0} + p_2 q_{\theta 2}}{p_0 + p_2}. \end{aligned} \tag{15}$$

Therefore, (14) may be rewritten as

$$\begin{cases} \widetilde{p}^{00} = \sum\limits_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^{M-k}, \\ \widetilde{p}^{10} = \sum\limits_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^{M-k}. \end{cases} \tag{16}$$

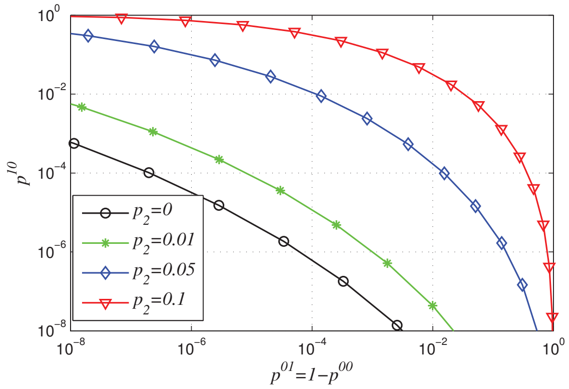

Fig. 1. Approximate p^{10} as a function of approximate p^{01} at equilibrium, for various $\nu \in [0,1]$ and $p_2 \in \{0, 0.01, 0.05, 0.1\}$.

from which one deduces the approximate values $\widetilde{X}_\theta^{M,k}$ of $\overline{X}_\theta^{M,k}$

$$\begin{cases} \widetilde{X}_0^{M,k} = \binom{M}{k} \left(\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^{M-k}, \\ \widetilde{X}_1^{M,k} = \binom{M}{k} \left(\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^{M-k}. \end{cases} \tag{17}$$

The quality of the approximation can be verified by checking whether there exists some value of ν that leads to both $\overline{p}^{00} \to 1$ (or $\overline{p}^{01} \to 0$) and $\overline{p}^{10} \to 0$. Some numerical comparisons between $\widetilde{X}_\theta^{M,k}$ and $\overline{X}_\theta^{M,k}$ will be presented in Section VI.

Consider here a toy example: fix $M = 20$ and the LODT is such that $q_{00} = 0.05$ and $q_{10} = 0.8$. The Byzantine nodes have the behavior of type *B2)* with $p_{02} = 1$ and $p_{12} = 0$, which corresponds to the most serious attack. Consider $p_2 \in \{0, 0.01, 0.05, 0.1\}$ and $p_0 = p_1 = (1-p_2)/2$ in all the cases, Figure 1 presents $\widetilde{p}^{10}$ as a function of $\widetilde{p}^{01}$, obtained for different values of $\nu \in [0,1]$. One observes that the Byzantine nodes have limited influence on the performance of the DFD algorithm, except when p_2 reaches 10%. Nevertheless, if the values of M and ν are properly chosen, both $\widetilde{p}^{01}$ and $\widetilde{p}^{10}$ can be kept relatively small even in presence of 10% of Byzantine nodes.

Figure 1 is also helpful to choose the value of ν in order to meet different performance requirements.

VI. Numerical Results

This section provides simulation results to illustrate the theoretical results presented in Section V. The results presented in Section VI-A are obtained considering nodes with an idealized displacement model. Some real databases are then considered in Section VI-B.

A. Idealized displacement model

Consider a DTN consisting of 1000 moving nodes, with their initial positions uniformly distributed over a unit square. Nodes randomly move within this square. Two nodes communicate only when their distance is less than their communication range r_0 at discrete time instants $k\Delta t$, $k = 1, 2 \ldots$.

$$\begin{cases} \overline{p}^{00} = \sum_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{00}\overline{p}^{00}+p_1 q_{01}\overline{p}^{10}+p_2 q_{02}}{p_0\overline{p}^{00}+p_1\overline{p}^{10}+p_2}\right)^k \left(1-\frac{p_0 q_{00}\overline{p}^{00}+p_1 q_{01}\overline{p}^{10}+p_2 q_{02}}{p_0\overline{p}^{00}+p_1\overline{p}^{10}+p_2}\right)^{M-k}, \\ \overline{p}^{10} = \sum_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{10}\overline{p}^{00}+p_1 q_{11}\overline{p}^{10}+p_2 q_{12}}{p_0\overline{p}^{00}+p_1\overline{p}^{10}+p_2}\right)^k \left(1-\frac{p_0 q_{10}\overline{p}^{00}+p_1 q_{11}\overline{p}^{10}+p_2 q_{12}}{p_0\overline{p}^{00}+p_1\overline{p}^{10}+p_2}\right)^{M-k}. \end{cases} \tag{14}$$

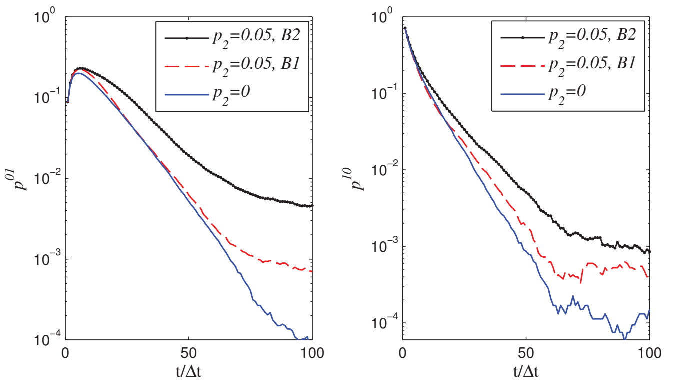

Fig. 2. Evolution of p^{01} (left) and p^{10} (right), considering an idealized displacement model.

One assumes an idealized displacement model: the location of each agent at time $(k+1)\,t$ is independent of its previous location at time $k\Delta t$. The value of r_0 can be chosen to adjust the inter-contact probability during a time interval of duration Δt. Here, the inter-contact probability is taken as 0.33.

Consider $N_\text{b} = 50$ Byzantine nodes and $N_\text{d} = 200$ nodes with defective sensors, which leads to $p_0 = 0.75$, $p_1 = 0.2$, and $p_2 = 0.05$. The characteristics of the LODT are $q_{00} = 0.05$, $q_{01} = 0.8$, and $q_{11} = 0.9$. Consider both types of Byzantine nodes: for the type B1), assume that $p_{02} = p_{12} = 1$; for the type B2), assume that $p_{02} = 1$ and $p_{12} = 0$. One also takes into account the situation where no Byzantine node is present, *i.e.*, $p_2 = 0$, in order to see the influence of Byzantine attack. In the latter case, one sets $N_\text{d} = 211$ so that the ratio of p_0 and p_1 are close in all the situations.

Figure 2 presents the evolution of p^{01} and p^{10} as functions of time, with $M = 15$ and $\nu = 0.4$. Recall that p^{01} is the proportion of normal nodes with good sensors that wrongly decide their sensors as defective and p^{10} is the proportion of normal nodes with defective sensors that wrongly decide their sensors as good. Compared with the situation where $p_2 = 0$, one observes that both p^{01} and p^{10} decreaser slower when the Byzantine nodes are present. As expected, the attack of type B2) impact more the agents compared than that of type B1). Figure 3 shows a good match between the distribution of $X_\theta^{M,k}$ obtained by the end of the simulation and the approximation of $X_\theta^{M,k}$ using (17). In order to have a good performance of the DFD algorithm, the distributions of $X_0^{M,k}$ and $X_1^{M,k}$ should be as separate as possible. The main influence of the Byzantine attack is that it makes the two distributions closer. Nevertheless, the DFD algorithm still behaves in a satisfying way if the parameter ν is properly chosen using (14): in the simulations both p^{01} and p^{10} can be made less then 1%.

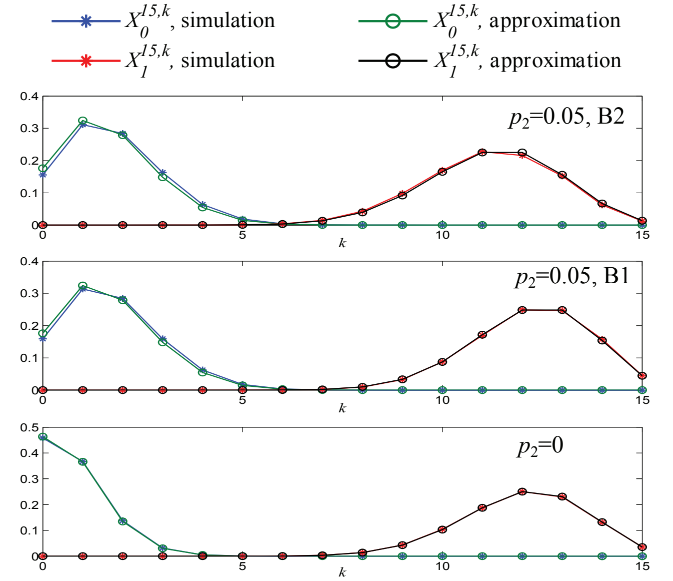

Fig. 3. Comparison of $X_\theta^{15,k}$ at the equilibrium, when 5% of nodes perform a byzantine attack of type B2 (top), of type B1 (middle), and when there are no Byzantine nodes (bottom).

B. Simulation with real databases

数据集 Haggle

In this section, the DFD algorithm is executed considering node inter-contact times taken from real databases provided by the Haggle Project [12] and by our own experiments conducted at the EuWin platform at University of Bologna. In the simulation, one is interested in the inter-contact trace, *i.e.*, which pair of agents have a meeting at which time. We use the following databases:

- *Infocom05*, in which $N = 41$, lasted 3 days.
- *Bologna16*, in which $N = 34$, during the break of a course (which lasts about 17 minutes).

For each database, 500 Monte-Carlo simulations are performed. In each simulation, one randomly choose N_b nodes as Byzantine nodes and N_d nodes as the ones with defective sensors. The results are then averaged over these simulations. In *Infocom05*, one sets $N_\text{b} = 2$ and $N_\text{d} = 10$. In *Bologna16*, one sets $N_\text{b} = 1$ and $N_\text{d} = 6$.

Consider the following parameters: $q_{00} = 0.05$, $q_{01} = 0.8$, $q_{11} = 0.9$, $p_{02} = 1$, $p_{12} = 0$, $M = 15$ and $\nu = 0.4$. At the top of Figure 4, the index of the active nodes (which have contact with the others) are presented at each time to show the frequency of the inter-contacts at different epochs. The evolution of p^{10} and p^{01} is plotted at the bottom of Figure 4. Interestingly, both p^{10} and p^{01} obtained by both databases decrease to 10^{-2} after a sufficient long time. The decreasing speed of p^{10} and p^{01} is highly related to the inter-contact rate

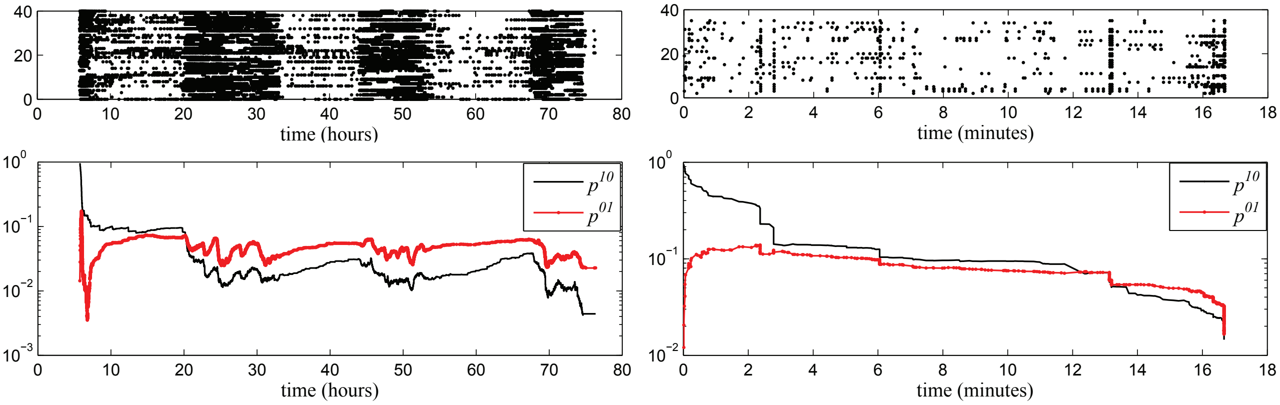

Fig. 4. Indexes of active nodes (having met another node) at different time (top) and evolution of p^{10} and p^{01}, obtained using the *Infocom05* database (left) and the *Bologna16* database (right).

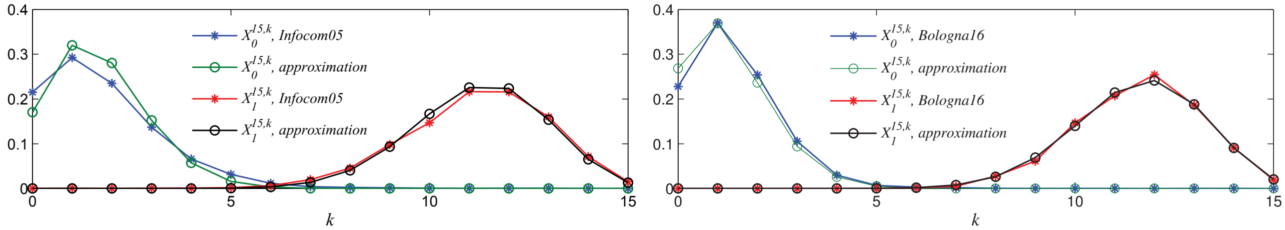

Fig. 5. Values of $X_\theta^{15,k}$ the end of the simulation, as well as the theoretical values at equilibrium obtained from (17), obtained using the *Infocom05* database (left) and the *Bologna16* database (right).

(reflected by the density of points in the sub-figures at the top): using *Infocom05*, variations are significant at beginning of working hours; using Bologna16, p^{10} and p^{01} decrease significantly in the end as all the students came back to the class.

Figure 5 represents the proportion of nodes in each state $X_\theta^{M,k}$ in the end of the simulation, obtained by using the databases *Infocom05* and *Bologna16*. The simulation results are compared with the approximation (17). One still finds that there is a good match by using the databases.

VII. Conclusion

This paper investigates the impact of Byzantine attacks on the performance of a distributed faulty node detection algorithm in the context of delay tolerant networks. The aim of the algorithm is to make each normal node estimate the status of its own sensors, whereas some Byzantine nodes attempt to mislead the behavior of the algorithm. The affect of Byzantine attack on the equilibrium is analyzed theoretically, which is helpful to adjust the algorithm parameters in order to ensure the robustness of the DFD algorithm. Both ideal movement model and real databases have been considered in the simulations to illustrate our results.

References

[1] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, “Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[2] P. R. Pereira, A. Casaca, J. J. Rodrigues, V. N. Soares, J. Triay, and C. Cervelló-Pastor, “From delay-tolerant networks to vehicular delay-tolerant networks,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1166–1182, 2012.

[3] W. Peng, F. Li, X. Zou, and J. Wu, “Behavioral malware detection in delay tolerant networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 53–63, 2014.

[4] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, “A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.

[5] E. Ayday and F. Fekri, “An iterative algorithm for trust management and adversary detection for delay-tolerant networks,” *IEEE Trans. on Mobile Computing*, vol. 11, no. 9, pp. 1514–1531, 2012.

[6] M. Abdelhakim, L. E. Lightfoot, J. Ren, and T. Li, “Distributed detection in mobile access wireless sensor networks under byzantine attacks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 950–959, 2014.

[7] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[8] A. Mahapatro and P. M. Khilar, “Fault diagnosis in wireless sensor networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2000–2026, 2013.

[9] W. Li, F. Bassi, D. Dardari, M. Kieffer, and G. Pasolini”, “Defective sensor identification for WSNs involving generic local outlier detection tests,” *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 29–48, 2016.

[10] W. Li, L. Galluccio, M. Kieffer, and F. Bassi, “Distributed faulty node detection in DTNs,” in *Proc. International Conference on Computer Communication and Networks*, 2016.

[11] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. Ni, “Recognizing exponential inter-contact time in vanets,” in *Proc. INFOCOM*, March 2010, pp. 1–5.

[12] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, “CRAWDAD dataset cambridge/haggle (v. 2009-05-29),” Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.